

BAB IV

IMPLEMENTASI

Dalam Bab ini, penulis menyusun implementasi dari metode yang digunakan.

4.1 Implementasi

4.1.1 Implementasi Metode Reverse Geocoding

Pada implementasi metode ini penulis membuat logika dalam menentukan lokasi *customer* menempati suatu meja yang telah di mapping lokasinya.

Pertama kali sistem akan mengecek *Longitude* dimana *customer* duduk lalu dicocokkan dengan data yang sudah diterapkan lalu dibandingkan dengan *Latitude* dimana *customer* duduk.

Contohnya, jika *customer* sedang berada pada *Longitude* 110.4109456 dan *Latitude* -7.7283165. Sistem akan mendeteksi *customer* berada di baris meja 10, 11 dan 12. Lalu sistem akan mengecek kembali dimana *Latitude* tempat *customer* berada. Sistem akan mendeteksi bahwa *customer* berada di barisan meja 10. Maka ditemukanlah lokasi *customer* berada.

Kelemahan pada sistem ini adalah ketika ada meja yang berdekatan memiliki kemungkinan lokasi akan tertukar mengingat pendeteksian lokasi koordinat pada *Google Map* sering melompat-lompat.

Berikut pengimplementasian metode dalam source code pada aplikasi Android :

```
//////////////////////////////////  
lat1 = -7.7283180;  
lat2 = -7.7283163;  
lat3 = -7.7283145;  
lat4 = -7.7283115;  
lat5 = -7.7283070;  
  
long1 = 110.4109377;  
long2 = 110.4109390;  
long3 = 110.4109416;  
long4 = 110.4109444;  
long5 = 110.4109471;  
long6 = 110.4109526;  
//////////////////////////////////  
poslong1 = false;  
poslong2 = false;  
poslong3 = false;  
poslong4 = false;  
poslong5 = false;  
poslong6 = false;  
poslat1 = false;  
poslat2 = false;  
poslat3 = false;  
poslat4 = false;  
poslat5 = false;
```

```

posk1confirm = false;
posk2confirm = false;
posk3confirm = false;
posk4confirm = false;
posk6confirm = false;
posk7confirm = false;
posk9confirm = false;
posk10confirm = false;
posk11confirm = false;
posk12confirm = false;
posk13confirm = false;

///

```

```

///// kursi 1 6 7 /////
else if (longitude > long2 && longitude < lon
    if (latitude > lat1 && latitude < lat2){
        posklconfirm = true;
        poslat1 = true;
    }
    else if (latitude > lat3 && latitude <
lat4){
        posk6confirm = true;
        poslat4 = true;
    }
    else if (latitude > lat4 && latitude <
lat5) {
        posk7confirm = true;
        poslat5 = true;
    }
    else {
        posk6confirm = false;
        posk7confirm = false;
        poslat4 = false;
        poslat5 = false;
    }
    poslong3 = true;
}

///// kursi 1 9 ///
else if (longitude > long3 && longitude < 1
    if (latitude > lat1 && latitude < lat2){
        posklconfirm = true;
        poslat1 = true;
    }
    else if (latitude > lat3 && latitude <
lat4){
        posk9confirm = true;
        poslat4 = true;
    }
    else{
        posk9confirm = false;
        poslat4 = false;
    }
    poslong4 = true;
}

```

```

//// kursi 1 10 11 12 //
else if (longitude > long4 && longitude < 1
    if (latitude > lat1 && latitude < lat2){
        posk1confirm = true;
        poslat1 = true;
    }
    else if (latitude > lat2 && latitude <
lat3){
        posk10confirm = true;
        poslat3 = true;
    }
    else if (latitude > lat3 && latitude <
lat4){
        posk11confirm = true;
        poslat4 = true;
    }
    else if (latitude > lat4 && latitude <
lat5){
        posk12confirm = true;
        poslat5 = true;
    }
    else {
        posk10confirm = false;
        posk11confirm = false;
        posk12confirm = false;
        poslat3 = false;
        poslat4 = false;
        poslat5 = false;
    }
    poslong5 = true;
}

// Kursi 13 //
else if (longitude > long5 && longitude <
long6){
    if (latitude > lat4 && latitude < lat5){
        posk13confirm = true;
        poslat5 = true;
    }
}

```

```
else {  
    poslong1 = false;  
    poslong2 = false;  
    poslong3 = false;  
    poslong4 = false;  
    poslong5 = false;  
    poslong6 = false;  
    poslat1 = false;  
    poslat2 = false;  
    poslat3 = false;  
    poslat4 = false;  
    poslat5 = false;  
    posk1confirm = false;  
    posk2confirm = false;  
    posk3confirm = false;  
    posk4confirm = false;  
    posk6confirm = false;  
    posk7confirm = false;  
    posk9confirm = false;  
    posk10confirm = false;  
    posk11confirm = false;  
    posk12confirm = false;  
    posk13confirm = false;  
}
```



```
if (posk1confirm)
{
    TvLokasiMeja.setText("1");
}
else if (posk2confirm)
{
    TvLokasiMeja.setText("2");
}
else if (posk3confirm)
{
    TvLokasiMeja.setText("3");
}
else if (posk4confirm)
{
    TvLokasiMeja.setText("4");
}
else if (posk6confirm)
{
    TvLokasiMeja.setText("6");
}
else if (posk7confirm)
{
    TvLokasiMeja.setText("7");
}
else if (posk9confirm)
{
    TvLokasiMeja.setText("9");
}
else if (posk10confirm)
{
    TvLokasiMeja.setText("10");
}
else if (posk11confirm)
{
    TvLokasiMeja.setText("11");
}
else if (posk12confirm)
{
    TvLokasiMeja.setText("12");
}
else if (posk13confirm) {
    TvLokasiMeja.setText("13");
}
else
{
    TvLokasiMeja.setText("Deteksi Gagal");
    TvLokasiMeja.setTextColor(Color.RED);
}
```

4.1.2 Implementasi Rancangan Antar Muka

4.2 Pengujian

Setelah tahapan desain dan tahapan implementasi selesai, maka dilakukan pengujian atau testing program aplikasi untuk melihat apakah sudah sesuai dengan apa yang direncanakan dan dibutuhkan baik pada *input* maupun *output* yang dihasilkan.

Untuk detailnya, pengujian ini menggunakan metode *black box testing* dan metode *white box testing*. Metode *black box testing* ini menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode pada program. *Black box testing* ini menguji fungsi dasar pada aplikasi yang dikembangkan sehingga dapat diketahui apakah sistem berlaku sesuai keinginan user.

Metode *white box testing* ini menguji perangkat apakah fungsi-fungsi *backend* sudah diterapkan dengan benar dan berfungsi dengan baik sehingga berjalan sesuai apa yang direncanakan dan diinginkan.

4.4.1 Black Box Testing

Table 1. Hasil Pengujian Halaman Login Costumer Aplikasi Android

No.	Skenario pengujian	<i>Test case</i>	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	Nama tidak diisi, deteksi meja gagal lalu ditekan login	Nama : kosong GPS : mati	Sistem akan menolak login		

2	Nama diisi, deteksi meja gagal, lalu ditekan login	Nama : Rafly GPS : mati	Sistem akan menolak login		
3	Nama diisi, deteksi meja otomatis berhasil	Nama : Rafly GPS : hidup Deteksi meja (otomatis) : meja 1	Sistem memperbolehkan <i>costumer</i> login. Data user tersimpan pada <i>database</i> .		
4	Nama diisi, deteksi meja manual	Nama : Rafly GPS : hidup Deteksi meja (manual) : meja 2	Sistem memperbolehkan <i>costumer</i> login. Data user tersimpan pada <i>database</i> .		
5	<i>Costumer</i> menekan “Lihat menu” pada menu login	<i>Costumer</i> menekan button “Lihat menu”	Sistem akan memperlihatkan semua menu yang tersedia		

Table 2. Hasil Pengujian Setelah *Costumer* Login ke Dalam Aplikasi Android

No.	Skenario pengujian	<i>Test case</i>	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	<i>Costumer</i> menekan menu profil	<i>Costumer</i> telah berhasil masuk ke dalam aplikasi	Nama, kode kursi & no kursi terdeteksi dan ditampilkan dengan benar pada menu profil		

Table 3. Hasil Pengujian Ketika *Costumer* Melakukan Order Pada Aplikasi Android

No.	Skenario pengujian	<i>Test case</i>	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	<i>Costumer</i> memilih makanan untuk diorder	Makanan : Mie Ayam Original	Setelah <i>costumer</i> memilih makanan, sistem akan melanjutkan ke halaman <i>checkout</i>		

2	<i>Customer</i> menambahkan jumlah makanan lalu menekan konfirmasi pesanan	Makanan : Mie Ayam Original Jumlah makanan : 2	Sistem akan menghitung jumlah harga dari 2 Mie Ayam Original		
3	<i>Customer</i> tidak memilih tipe makanan lalu menekan konfirmasi pesanan	<i>Radio Button</i> tipe makanan tidak terpilih satupun	Sistem akan menolak melakukan perhitungan total pesanan sebelum <i>customer</i> memilih tipe makanan		
4	<i>Customer</i> memilih satu atau lebih topping lalu menekan konfirmasi pesanan	Topping : Extra Sawi Tombol konfirmasi pesanan ditekan	Sistem akan menghitung total harga dari jumlah makanan dan harga topping yang dipilih		

5	<i>Customer</i> melakukan order	<i>Customer</i> menekan tombol order	Sistem akan menyimpan order kedalam <i>database</i> dan melanjutkan ke halaman notifikasi dan menampilkan order yang dibuat dan status pesanan.		
---	---------------------------------	--------------------------------------	---	--	--

Table 4. Hasil Pengujian Halaman Login Admin pada Web Admin

No.	Skenario pengujian	<i>Test case</i>	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	<i>Username</i> dan <i>Password</i> tidak diisi lalu login	<i>Username</i> : kosong <i>Password</i> : kosong	Sistem akan menolak untuk login		
2	<i>Username</i> tidak diisi dan <i>Password</i> diisi lalu login	<i>Username</i> : kosong <i>Password</i> : admin123	Sistem akan menolak untuk login.		
3	<i>Username</i> diisi dan <i>Password</i>	<i>Username</i> : Admin	Sistem akan menolak untuk login.		

	tidak diisi lalu login	<i>Password</i> : kosong			
4	<i>Username</i> diisi dan <i>Password</i> diisi dengan benar lalu login	<i>Username</i> : Admin <i>Password</i> : Admin123	Sistem akan memperbolehkan admin untuk masuk		

Table 5. Hasil Pengujian Halaman *Dashboard* Admin pada Web Admin

No.	Skenario pengujian	<i>Test case</i>	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	Website melakukan <i>auto refresh</i> tiap 10 detik	Admin telah login ke dalam dashboard	Sistem melakukan <i>auto refresh</i> setiap 10 detik		
2	Admin memberhentikan <i>auto refresh</i> dengan menekan tombol <i>stop refresh</i>	Admin menekan tombol <i>stop refresh</i>	Sistem memberhentikan <i>auto refresh</i> dan web tidak akan <i>refresh</i> otomatis		
3	Saat ada order masuk, web akan	Web mengeluarkan suara	Web mengeluarkan suara notifikasi		

	mengeluarkan suara	berulang ulang sampai web ter refresh	menandakan pesanan masuk		
4	Saat ada order masuk, order akan ditampilkan	Order ditampilkan pada tabel pesanan baru dan pada tabel lokasi meja	Pada tabel pesanan baru, order ditampilkan secara sederhana. Pada tabel lokasi meja, kotak tabel akan berwarna hijau tergantung pada no meja berapa orderan tersebut		
5	Admin menekan order pada tabel pesanan baru	Admin menekan order pada pesanan baru	Tabel detail order akan menampilkan isi dari semua order berdasarkan no meja pada order tersebut		
6	Admin menekan no meja yang	Admin menekan no	Tabel detail order akan		

	berwarna hijau pada tabel lokasi meja	meja yang berwarna hijau pada tabel lokasi meja.	menampilkan isi dari semua order berdasarkan no meja pada order tersebut.		
7	Admin menekan no meja berwarna putih pada tabel lokasi meja	Admin menekan no meja yang berwarna putih pada tabel lokasi meja	Tabel detail order tidak akan menampilkan order.		
8	Admin menekan tombol selesai pada tabel detail order kolom konfirmasi pesanan	Tombol selesai pada tabel detail order kolom konfirmasi pesanan ditekan	Status pesanan akan berubah menjadi selesai dan menghilang pada tabel detail order menandakan pesanan telah selesai		

Table 6. Hasil Pengujian Integrasi Web Service ke Aplikasi Android

No.	Skenario pengujian	<i>Test case</i>	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	Admin menyelesaikan pesanan lewat web	Tombol selesai ditekan oleh admin pada web	Status pesanan pada tab notifikasi pada aplikasi android berubah menjadi selesai menandakan pesanan telah selesai dan akan diantar ke <i>costumer</i>		

4.4.2 White Box Testing

Table 7. Hasil Pengujian Backend pada Aplikasi Android

No.	Skenario pengujian	<i>Test case</i>	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	Logika berhasil menemukan lokasi <i>costumer</i> berada berdasarkan koordinat yang diterapkan	Auto deteksi berjalan dan menampilkan lokasi meja	Sistem berhasil menemukan lokasi user dengan benar		
2	Order telah dibuat oleh <i>customer</i>	<i>Costumer</i> membuat order	Order tersimpan pada database di tabel orders		
3	<i>Costumer</i> login	<i>Costumer</i> login dengan berhasil	Data <i>Costumer</i> tersimpan pada database di tabel guest		
4	Data lokasi <i>Costumer</i> ditampilkan	Data <i>Costumer</i> ditampilkan	Data <i>Costumer</i> ditampilkan pada tab profile		

5	Data order <i>Costumer</i> ditampilkan	Data order <i>Costumer</i> ditampilkan	Data semua order yang dibuat oleh user ditampilkan pada tab notifikasi		
---	--	--	---	--	--

No.	Skenario pengujian	<i>Test case</i>	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	Dashboard dapat menyesuaikan ukuran jika dibuka melalui <i>smartphone</i>	Web dibuka pada <i>smartphone</i>	Layout web menjadi responsif jika dibuka melalui <i>smartphone</i>		
2	Web cepat dalam menampilkan dashboard	Web memuat dashboard	Web memuat dashboard dengan cepat		

4.3 Pendukung (*support*) atau Pemeliharaan (*maintenance*)

Dalam perancangan dan pembuatan software, software tidak selalu sempurna dan berjalan semestinya, mungkin masih ada beberapa *error* dan *bug* yang berada dalam aplikasi yang tidak terdeteksi pada tahap pengujian.

Dan pada tahapan pemeliharaan, perangkat lunak harus beradaptasi dengan lingkungan baru, sehingga pemeliharaan harus dilakukan agar aplikasi dapat berjalan semestinya, dengan tidak perlu membuat aplikasi baru, namun hanya perlu untuk dilakukan pengembangan pada aplikasi yang sudah ada.

DAFTAR PUSTAKA

- [1] “Does the strength of GPS weaken in rural areas?,” *Geographic Information Systems Stack Exchange*. <https://gis.stackexchange.com/questions/77929/does-the-strength-of-gps-weaken-in-rural-areas> (accessed Jan. 04, 2022).
- [2] M. T. Prihandoyo, “Unified Modeling Language (UML) Model Untuk Pengembangan Sistem Informasi Akademik Berbasis Web,” *J. Inform.*, p. 4, 2018.
- [3] D. Intern, “Apa itu UML? Beserta Pengertian dan Contohnya,” *Dicoding Blog*, May 11, 2021. <https://www.dicoding.com/blog/apa-itu-uml/> (accessed Dec. 24, 2021).